## IN THE SPECIFICATION

Please amend paragraph number [00045] on page 8 as follows:

**[00045]** Referring now to FIG. 2A, FIG. 2A further illustrates the host computer 200, as depicted in FIG. 1, in accordance with the further embodiment of the present invention. As depicted, the host computer 200 includes a CPU 202, a user interface 204, a communications interface 206, ~~as well as a~~ memory 212 as well as a network interconnectivity 246. However, the host computer 200 is implemented utilizing a Bluetooth™ controller 300, which performs device authentication initialization utilizing, in one embodiment, audio device identification information, as the authentication initialization information. In one embodiment, the audio device identification information can be utilized as both an initialization key, as well as an authentication key.

Please amend paragraph number [00046] on page 8 as follows:

**[00046]** Although the host computer 200 is illustrated using the Bluetooth™ controller 300, those skilled in the art will recognize that each of the Bluetooth™ enabled devices, such as wireless slave devices 180 (180-1, . . ., 180-N), also include a Bluetooth™ controller 300 and Bluetooth™ radio ~~310~~210, as depicted in FIG. 2B. Accordingly, the Bluetooth™ enabled slave devices 180 can be utilized by the host device in order to provide communication without the need for interconnected wires between the host computer and the Bluetooth™ enabled slave devices.

Please amend paragraph number [00054] on page 13 as follows:

**[00054]** Referring now to FIG. 3, FIG. 3 further illustrates the Bluetooth™ controller 300, as depicted in FIGS. 2A and 2B, in accordance with a further embodiment of the present invention. As depicted, the Bluetooth™ controller 300 includes an analog portion as the Bluetooth™ radio 310, which is utilized to communicate with the various Bluetooth™ enabled devices, as well as the digital portion, the link controller (Bluetooth™ baseband controller) 500. Accordingly, the Bluetooth™ baseband controller 500 performs the digital signal processing functions of the Bluetooth™ controller 300 using various signal processing hardware. The Bluetooth™ controller 300 further includes a CPU (central processing unit) core 360, which is optional, but is however beneficial, in order to avoid interface problems with a personal computer CPU. The CPU core ~~402~~ 360 is further utilized to interface between external interfaces ~~404~~402, which provide a host interface 400 to the various additional devices of the host computer 200.

Please amend paragraph number [00057] on page 14 as follows:

**[00057]** Referring now to FIG. 4, FIG. 4 ~~depicts~~ further illustrates link manager 350 and host interface 400, as depicted in FIG. 3, in accordance with one embodiment of the present invention. The host interface (HIF) 400 includes hardware and software, which interfaces the Bluetooth™ controller 300 to a host, such as for example, host device 200, as depicted in FIG. 2. The primary function of HIF 400 is to provide services of the lower layers (via the link manager) in a format suitable for the host device. Representatively, link manager 350 may include universal synchronous receiver/transmitter (UART) 352, random access memory (RAM) 354, serial PCM interface 356, which is coupled to analog-to-digital converter/digital-to-analog converter (ADC/DAC) 358. In the embodiment illustrated, UART 352, RAM 354 and serial pulse code modulation (PCM) interface 356 of link manager 358 are coupled to CPU core via bus 404.

Please amend paragraph number [00058] on page 14 as follows:

**[00058]** Accordingly, the various external interfaces are depicted in FIG. 4, which may include for example, UART 410, PC card interface 430, PCI (peripheral component interconnect) interface 440, low pin out pin count interface 450, USB interface 460 and dual port memory interface 470. These devices may implement and support ~~PC~~ PC-card 480, card bus 442, ~~RS~~ RS232 connector 412, PC mother board integration 490, as well as USB slave 462. As such, the Bluetooth™ controller firmware implements the baseband link management protocol. The drivers control the radio 310 using the Bluetooth™ host controller interface 400, which is accessed through an appropriate transport interface, as described in further detail below.

Please amend paragraph number [00059] on page 14 as follows:

**[00059]** Referring now to FIG. 5, FIG. 5 further illustrates the Bluetooth™ baseband controller, as depicted in FIG. 4. The Bluetooth™ baseband controller 500 includes a processor 360 along with memory arbiter 540, SRAM sequencer 550, flash memory 560 and SRAM 570. However, in contrast to conventional baseband controllers, the Bluetooth™ baseband controller 500 includes audio authentication unit 530, which baseband ~~220~~ 520 utilizes in order to request audio (voice stream) device identification information (code) ~~362~~ 562 assigned to the device by a user, and stored within, for example, flash memory 560. This audio device identification information replaces the initialization key utilized by Bluetooth™ security during first time audio link initialization. As a result, the user is not required to enter device ID codes and PIN numbers via MMI. Accordingly, the Bluetooth™ baseband 520 includes firmware for performing the audio device initialization and

authentication functionality of link manager driver procedures, as depicted in FIGS. 2A and 2B, at the baseband level.

Please amend paragraph number [00064] on page 15 as follows:

**[00064]** Referring now to FIG. 6, FIG. 6 depicts the Bluetooth™ software stack 600. Representatively, Bluetooth™ software stack 600 includes standard user mode API 610, Bluetooth™ bus driver interface 620 and transport bus interface 680. As illustrated, the core of the Bluetooth™ software stack 600 essentially includes the Bluetooth™ bus driver 630, the Bluetooth™ host control interface (HCI) 640 and the Bluetooth™ host controller driver 650. Accordingly, each function class typically includes a client driver that is loaded by the Bluetooth™ bus driver. These client drivers utilize the Bluetooth™ bus driver interface 620 to communicate with the Bluetooth™ bus driver 630 for data and control transfer purposes.

Please amend paragraph number [00065] on page 26 as follows:

**[00065]** However, in contrast to conventional Bluetooth™ software stacks, Bluetooth™ software stack 600 includes a Bluetooth™ or link manager control driver 660, which is utilized to implement the audio device authentication initialization 670, as described herein. However, those skilled in the art will recognize that the audio authentication methods described herein may be implemented anywhere within the Bluetooth™ bus driver interface 620 software stack 600, as well as within the Bluetooth™ controller firmware, depending on the desired implementation specific details of the system designer.

Please amend paragraph number [00076] on page 18 as follows:

**[00076]** Referring now to FIG. 9, FIG. 9 depicts an additional method ~~708~~ 710 for storing a received audio device identification information of process block 708, as depicted in FIG. 8. At process block 712, the received audio device identification information is compressed. Once compression is complete, at process block 714, the host device 200 generates a hash value of the compressed audio device identification information to form the authentication initialization token of the device 180. As such, in the embodiment described, the hash value of the stored authentication initialization token will be compared with the hash value of the received authentication initialization token in order to determine whether matching audio device identification information is detected in order to complete device authentication initialization.

Please amend paragraph number [00080] on page 20 as follows:

**[00080]** Referring now to FIG. 12, FIG. 12 depicts an additional method 752 for authenticating the detected device using the requested device authentication initialization information of process block 750, as depicted in FIG. 7. At process block ~~752~~754, the host device receives audio device identification information as the requested audio authentication initialization information of the detected device. Once received, the host computer compresses the received audio device identification information at process block 756. At process block 758, a hash value of the compressed audio device identification information is generated in order to form a requested device authentication initialization token.

Please amend paragraph number [00085] on page 21 as follows:

**[00085]** Referring now to FIG. 16, FIG. 16 depicts a method 802 for audio authentication set-up of a wireless slave device, utilizing for example, device audio authentication set-up procedures 190. At process block 804, it is determined whether a device authentication set-up request is received. In one embodiment, the request is automatically received during genesis boot of the wireless device 180. Next, at process block 806, the wireless device requests from the user a spoken or voice stream (audio) device name or device identification code (identification information). Next, at process block 808, it is determined whether the audio device identification information is received from the user. Once received, at process block 810, the wireless device 180 stores the audio device identification information as an authentication initialization token of the device 180. In one embodiment, as described above, the audio device identification information replaces an initialization key utilized according to the Bluetooth™ System Specification during initialization set-up of a communication link between a slave device and a host device.

Please amend paragraph number [00087] on page 22 as follows:

**[00087]** Referring now to FIG. 17, FIG. 17 depicts an additional method 822 for determining whether an audio authentication initialization request is received. At process block 824, it is determined whether device identification information is requested from the wireless device 180. As described above, this process generally involves device authentication and a challenge response scheme to illustrate on the behalf of the wireless device, knowledge of an authentication key stored within the host device during device initialization. Next, at process block 826, it is determined whether the requested device identification information is authenticated by the host device 200. When audio authentication information is authenticated, an audio link is established between the host device 200 and the detected wireless device 180 , as indicated at process block 852, as depicted in FIG. 15. Otherwise, at process ~~clock~~ block 828, the wireless device receives a

request for audio authentication initialization information from the host device 200. In one embodiment, method 822 is performed utilizing the authentication reply procedures 196.

Please amend paragraph number [00089] on page 22 as follows:

**[00089]** Finally, referring to FIG. 19, FIG. 19 depicts an alternate method 842 for providing audio authentication initialization information to a requesting host device 200 at process block 832 (FIG. 15). Accordingly, at process block ~~834~~844, the wireless device selects a stored audio authentication initialization token. As described in an embodiment above, the audio authentication initialization token is provided by the user via the device audio authentication set-up procedures and functions as a substitute initialization key. In one embodiment, the audio authentication initialization token is a spoken device name or device identification code that is assigned by the device user during genesis boot of the device. Alternatively, the audio device identification information may be factory installed, thereby limiting the user to providing of the factory installed information to the host device for the authentication initialization process. Next, at process block 846, the devices transmits the audio authentication token to the requesting host device.